**WORKSHOP - 8 (Self Checkout)**

**Group Solution**

A diagram of a process flowchart

Description automatically generated

DATA STRUCTURES

- item\_info

- item\_list

- item\_price

- item\_status

- np\_subTotal = 0.00

- p\_subTotal = 0.00

- subTotal = 0.00

- tax = 0.00

**BLACKBOX PROCESS**

1. PAYMENT PROCESSOR : SUB - PROCESS

2. DATA LOOKUP: SUB - PROCESS

**MAIN PROCESS**

1. START

2. DISPLAY:

Is this item Scanable?

(Yes) (No)

3. If the user chooses "Yes":

- Call <NON PERISHABLE : SUB - PROCESS>

4. If the user chooses "No":

- Call <PERISHABLE: SUB - PROCESS>

5. Display:

Are there more items?

Yes No

6. Did the user choose "Yes"?

(YES):

Return to step #2

(NO):

Proceed to the next step.

7. Call <PAYMENT: SUB - PROCESS>

**NON PERISHABLE: SUB - PROCESS**

1. START

2. n = 1

3. DISPLAY:

Please Scan your items

4. Call <DATA LOOKUP : SUB - PROCESS>

5. Get item\_info

6. DECALRE VARIABLES:

- product\_cost = 0.00 (Received from DATA LOOKUP)

7. item\_price[n] += product\_cost

8. item\_list[n] += item\_info

9. item\_status[n] += 'Non Perishable'

10. n += 1

11. END

**PERISHABLE: SUB - PROCESS**

1. START

2. n = 1

3. DISPLAY:

Enter Item Code:

4. Call <DATA LOOKUP: SUB - PROCESS>

5. Get item\_info

6. DECALRE VARIABLES:

- item\_weight = 0.00

- product\_cost = 0.00 (Received from DATA LOOKUP)

7. DISPLAY:

DOES THE ITEM REQUIRE WEIGHING OR QUANTITY

(WEIGHING) (QUANTITY)

8. Did the user choose "WEIGHING"?

(YES):

(a) \*Weigh Item

(b) Get item\_weight

(c) item\_price[n] += item\_weight \* product\_cost

(d) item\_list[n] += item\_info

(e) item\_status[n] = 'Non Perishable'

(f) n += 1

(NO):

(a) DISPLAY:

"ENTER THE ITEM QUANTITY"

(b) Get item\_weight

(c) item\_price[n] += item\_weight \* product\_cost

(d) item\_list[n] += item\_info

(e) item\_status[n] = 'Non Perishable'

(f) n += 1

9. END

**PAYMENT: SUB - PROCESS**

1. Start

2. n=1

3. Receive List of perishable and non-perishable items.

4. Scan all the items.

5. Check is item Non-perishable ?

-YES :

np\_subTotal += itemPrice[n]

-NO:

p\_Subtotoal += itemPrice[n]

6. Is there any item left in list to scan?

-YES:

# Go back to step 3

-NO:

Proceed to the next step

7. subTotal = np\_Subtotal + p\_Subtotal

tax = np\_Subtotal \* TAX // np\_SubTotal \* 0.13

total = subTotal + tax

8. Display :

subTotal = [subtotal]

tax = [tax]

Total = {subTotal + total]

9. Display:

Please choose a Payment Option

- Credit Card - Debit Card

10. Which payment option does the user chose ?

11. If user chose "Credit Card"

YES :

Call Payment Processor : BLACKBOX PROCESS

12. If user chose "Debit Card"

Yes :

Call Payment Processor : BLACKBOX PROCESS

13. Check that was payment successful?

Yes:

Display : Payment Successful

No

Display :

"ERROR: PAYMENT CANCELLED"

#Go back to Step 1

14. Do you wanna print receipt?

Yes:

Call RECEIPT: SUB-PROCESS

No:

Display: Thank You For Shopping :)

15. END

**RECEIPT: SUB - PROCESS**

1. START

2. Display:

Product Name Product Price

item\_list[] item\_price[]

3. Display:

SUBTOTAL - [subTotal]

Taxes - [tax]

Total- [total]

4. Display :

Date: DD/MM/YYYY

Time - 00:00:00

5. Display : THANK YOU FOR SHOPPING

6. END

**TEST SCENARIO - 1**

1. The user has three items to buy

- 1 lb of perishable item ($5.5/lb)

- 6 cans of non-perishable item ($3.00/Can)

- 2 cartons of non-perishable item ($7.00/carton)

2. User selects "NO" for the scannable option

3. item 1 cost gets calculated (5.5)

4. User selects "YES" for the scannable option

5. item 2 cost gets calculated (18.00)

6. item 4 cost gets calculated (14.00)

7. tax gets calculated ( [18 + 14] X 0.13 = 4.16 )

8. User pays $41.66 by credit card

**TEST SCENARIO - 2**

1. The user selects one item to by

- 5 lb avocado ($3.2/lb)

2. User selects "NO" for scannable option

3. item cost gets calculated (16.00)

4. No tax is applied

5. User pays $16.00 by debit card

**Computational thinking**

Decomposition:

The first aspect is we interface design, which includes the arrangement and features for scanning products, choosing payment methods, and generating receipts.

Item processing includes creating algorithms for scanning barcodes, identifying products, retrieving data from a database, and managing discounts, promotions, and inventory.

Implementing secure systems is key in payment processing to handle different payment methods such as cash, credit cards, and mobile payments, along with strong security measures. Furthermore, the process of generating receipts involves developing algorithms that can create detailed receipts with correct product information and prices. We use Error handling mechanisms which are crucial for dealing with incorrect inputs or transaction errors.